(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification⁷: G06F 15/16, H04L 12/00

(21) International Application Number: PCT/US01/40256

(22) International Filing Date: 6 March 2001 (06.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/187,342   6 March 2000 (06.03.2000)   US
09/800,169   5 March 2001 (05.03.2001)   US

(71) Applicant: i2 TECHNOLOGIES, INC. [US/US]; One i2 Place, 11701 Luna Road, Dallas, TX 75234 (US).

(72) Inventors: DUNN, Craig, L.: 12311 Old Stage Trail, Austin, TX 78750 (US). MCGANN, Conor: 7740 Yaupon Drive, Austin, TX 78759 (US).

(74) Agent: CHRISTOPHER, W. KENNERLY; Baker Botts L.L.P, 2001 Ross Ave., STE. 600, Dallas, TX 75201-2980 (US).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European

(54) Title: MESSAGING SYSTEM FOR COMPUTERS
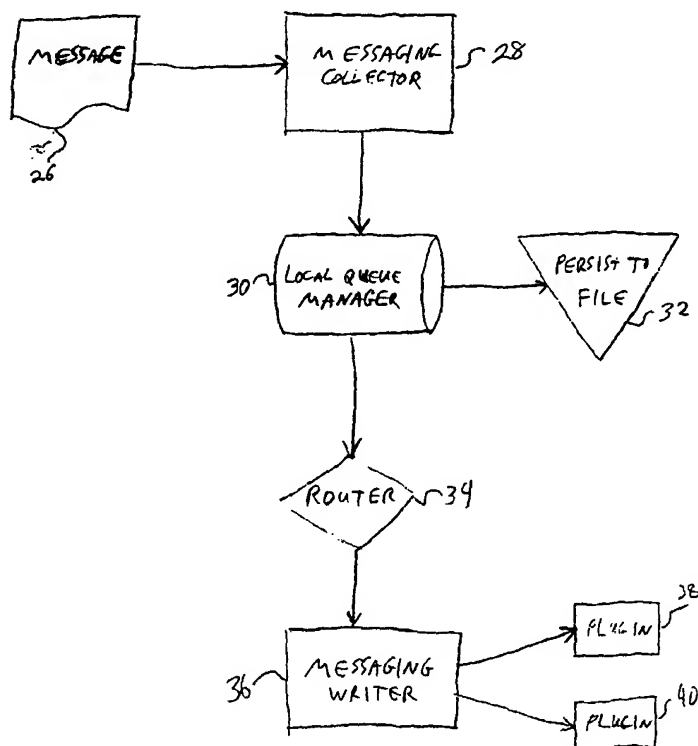
(57) Abstract: In accordance with the present invention a messaging system utilizes a local queue manager (30, fig.2) to receive messages (26, fig.2) intended for other processes. Messages received by the queue manager (30, fig.2) are stored onto a local persistent storage device, and a process sending the message has completed the sending action. The local queue (30, fig.2) manager then sends the message to an appropriate recipient. When the message has been received and confirmed, the recipient removes the message from the persistent storage device. If a hardware or software failure occurs, the message is stored and can be re-sent after the failure is corrected.

WO 01/67263 A1

patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:
— *with international search report*

- 1 -

MESSAGING SYSTEM FOR COMPUTERS

Technical Field

The present invention relates generally to computer systems, and more specifically to message handling between processes and a distributed computer system.

5   Description of the Prior Art

As electronic computer systems and the processes executed on the systems become more complex, it is common to distribute processing in some manner. Further, as computing becomes more distributed over communication networks, including worldwide networks such as the internet, distributed processing becomes a desirable
10   approach to computing.

When processing on problems becomes distributed, whether locally or remotely, communication between various processes becomes more important. These processes are sometimes referred to as modules, and must communicate with each other to perform the overall processing tasks. One common technique for communication between
15   remotely located processes is the use of message passed between them. By using messages, it is not necessary that the processes be tightly coupled, and each process can be optimized to perform its function.

Available systems and methods for passing messages between distributed processes are not reliable enough for some applications. In certain critical applications,
20   failure of a software or hardware module could interfere with transfer of a message between processes. If a message is lost or garbled, it may be difficult and impossible for the processes involved to accurately recover from the failure. Also, many such systems are not fast enough to ensure adequate response times as seen by users.

It would be desirable for an improved messaging system and method to reliably
25   handle messages, and insure that messages between processes are properly routed and confirmed as received by the intended recipient process.

Summary of the Invention

In accordance with the present invention a messaging system utilizes a local queue manager to receive messages intended for other processes. Messages received by the queue manager are stored onto a local persistent storage device, and a process sending the message has completed the sending action. The local queue manager then sends the message to an appropriate recipient. When the message has been received and confirmed, the recipient removes the message from the persistent storage device. If a hardware or software failure occurs, the message is stored and can be re-sent after the failure is corrected.

Brief Description of the Drawings

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself however, as well as a preferred mode of use, further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Figure 1 and 2 are block diagrams illustrating a messaging system in accordance with the present invention; and

Figure 3 is a block diagram illustrating handling of multiple messages between processes.

Description of the Preferred Embodiment

As will be appreciated by those skilled in the art, the system described below can be implemented on nearly any system that passes messages between processes. Preferably, messages are handled asynchronously. This means that once the message is sent from the sending process, timing of receipt of the message by the receiving processes is generally not critical. In the description below, it is assumed that the sending and receiving processes are not tightly coupled, and that a message once sent need not have delivery confirmed.

- 3 -

Referring to Figure 1, a sending process 20 needs to send a message, as generally known in the art, to receiving process 22. This message can be, for example, a remote database access query, a status request, notification of an event occurring within sending process 20, or any other subject matter suitable for such remote messaging.

5   The message is sent from sending process 20 to message handling system 24, which eventually routes the message to receiving process 22. The system and method described herein are particularly applicable to the design and operation of the message handling system 24.

Referring to Figure 2, message 26 is sent to messaging collector 28. Messaging
10   collector 28 serves as the interface to the outside world for message handling system 24. Messaging handling collector 28 is preferably an object that is invoked by sending process 20 in order to handle the message to be sent. In a preferred embodiment, messaging collector 28 is invoked by a Java class call, and invokes a single method for each method passed. Those skilled in the art will recognize that messaging collector 28 may be
15   implemented differently to perform the same functions as described herein.

Messaging collector 28 immediately passes the message off to local queue manager 30. Because local queue manager 30 is local to sending process 20, this happens very quickly and results in minimal latencies. Once messaging collector 28 has passed the message off to local queue manager 30, sending process 20 has completed
20   sending message 26, and may return to other functions.

This approach causes message handling to be asynchronous as possible. From this point forward, the message handling system 24 will handle the responsibility for delivering message 26.

The messaging collector method 28 preferably returns a status code to sending
25   process 20 indicating success or failure for accepting the message. This is based upon a similar success or failure status received from the local queue manager. Once an indication of success is returned, the sending process 20 assumes that the message will be reliably delivered

Local queue manager 30 accepts messages as quickly as possible from messaging collector 28, and prepares them for continued transmission. Preferably, multiple messaging collector objects 28 communicate with local queue manager 30. Thus, local queue manager 30 can receive numerous messages from different messaging

5   collector methods 28, which are in turn evoked by one or more sending processes. When the message is received by the main thread of local queue manager 30, the message is queued to a static internal class vector. Also, preferably, it is persisted to a local file system 32, where it is stored until the message is delivered. Until removed by the receiver, the message remains on the local file system so that it can be retrieved and

10   resent in case of a hardware or software failure somewhere along the line. This increased tolerance of hardware and network failures occurs at a cost of slightly increased overall latency. However, for applications that need to ensure message delivery, this tradeoff is a good one.

Local queue manager 30 also includes a separate thread to manage outbound

15   traffic. This thread pulls messages off the internal queue, preferably in FIFO order, and makes a call to a configured routing object 34. The call includes at least a message destination taken from the message itself, and router 34 returns a destination server to local queue manager 30. Once the destination server has been obtained, the message is passed by a standard messaging system to such server.

20   The destination server identified by router 34 can be implemented in any identifying manner compatible with the remainder messaging system. As described below, the message is passed to a messaging writer 36, the identification and location of which is stored in a registry or similar routing table. The routing algorithms utilized by router 34 can be any that are appropriate to the specific implementation, and can be easily changed at

25   any time due to the separation between router 34 and local queue manager 30. This allows the routing algorithm to be changed or enhanced when, for example, the overall system is expanded and the messaging load increases.

Message writer 36 is a module that receives messages and routes them to a specific associated process. In the example in Figure 2, plug in processes 38, 40 are

30   associated with message writer 36, and receive messages from it.

- 5 -

Within messaging writer 36, a primary thread preferably receives incoming messages, and places them on to a background thread queue. The background thread determines the type of message and routes it to a specific queue for the associated plug in 38, 40. These queue threads then simply pass the messages to the plug ins 38, 40 for
5   final receipt and process.

Once the message writer 36 has delivered the message, it removes the message form the local queue, where it had been placed by the local queue manager 30. Until message writer 36 has delivered the message, it remains on the persistent storage device, from where it can be accessed later if need be. Thus, in the event of a failure, the
10   message is saved in a reliable location. Once the failure has been corrected, the message can be resent and receipt insured by plug in 38, 40, the intended recipient. This persistent storage of messages on a local file server enables recovery from failures to be performed in a simple manner, without requiring the messages to be re-generated by sending processes 20, a process which may not be possible due to the nature of the
15   sending process.

The latency of messages handled by such system can be controlled to be acceptably small. Initially, the local latency from sending, or client, processes 20, are very small, due to the local nature of messaging collector 28 and local queue manager 30. End-to-end latency, defined as the time it takes a process to send a message to plug-in
20   38 or 40, depends on the detailed design of message handling system 24 and the current load the system is handling. Message handling system 24 can be increased in size if a lower end-to-end latency is required, and still function in the same manner as described above.

Referring to Figure 3, in block diagram of a more complete message handling
25   system 24 is illustrated. Multiple messages 51-56 are generated by multiple sending processes (not shown). Each message 51-56 is sent to its associated messaging collector 61-66. In this example, three local queue managers 71, 72, 73 are used, and each message is sent to an associated local queue manager as shown. Each local queue manager then persists, or stores in a long-term manner, each message to a local file
30   system 74, 75, 76. Each local queue manager 71-73 invokes an associated router

method 81, 82, 83, which handles routing of messages to appropriate messaging writers 91, 92, 93.

Message writers 91-93 determine the underlying type of message, and route them to an appropriate plug in as shown. As described previously, the appropriate plug in
5   converts the message to the appropriate format, and forwards its to the receiving process 22.

As can be seen from Figure 3, message handling system 24 is extremely fast, and can be easily extended. The message handling system is not concerned with the type of underlying message, but transfers all messages in the same way. The underlying type of
10   the message is only a concern when the message reaches the appropriate messaging router, which sends it to the appropriate plug in for reconversion back to the original message type. All message types are treated in the same manner by the message transport system.

Additional local queue managers and messaging writers can be added at any time.
15   As described above, routing algorithms are easily modified at any stage to perform routing that is either optimal, or good enough for the specific implementation.

The system described above is simple in operation and economical in design. It allows messages to be reliably transmitted and received, even in the event of a system failure, because the message is not marked as received by the local queue manager until
20   it is stored onto a reliable long term storage system. Because the message remains in persistent storage until removed after receipt, it can always be re-sent after a failure.

Because the transport system does not care about the type of the underlying message, it can be used for messages of any type. The message collector 28 formats any type of message into a standard type, and it is returned to its original type in the
25   appropriate plug in selected by the message writer 36. Thus, multiple new message types can be added to the system without impacting the message handling performed on the messages. Whenever a new message type is defined, the message collector method 28 is updated, and a new plug in is prepared to convert those messages back into their original type. No additional modifications are required.

While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.

5

Claims

1.   A messaging system for a computer system, comprising:

a sending process, wherein the sending process generates a message to be sent;

a receiving process, wherein the message is to be sent to the receiving process;

a local queue manager in communication with the sending and receiving processes; and

a persistent storage device in communication with the local queue manager, and adapted to reliably store messages until they are removed;

wherein the local queue manager stores each message received from the sending process to the persistent storage device, and wherein the receiving process removes each message from the persistent storage device after each message is received.

2.   The system of Claim 1, further including a message writer in communication with the local queue manager and the receiving process, wherein the message writer removes each received message from the persistent storage device after sending it to the receiving process.

3.   The system of Claim 1, further comprising a message collector in communication with the sending process and the local queue manager, wherein the message collector receives a message from the sending process, and formats the message into a standard format for transport to the local queue manager and storage on the persistent storage device.

4.   The system of Claim 3, further comprising:

at least one process, associated with the receiving process, for converting received message back into their original format from the standard format.

5.   A method for sending messages within a computer system, comprising the steps of:

generating a message within a sending process;

storing a copy of the message within a persistent storage device;

5   sending the message to a receiving process; and

removing the stored copy after the message has been received by the receiving process.

6.   The method of Claim 5, further comprising the step of:

10   after storing the copy of the message on the persistent storage device, sending an acknowledgement thereof to the sending process.

7.   The method of Claim 5, further comprising the steps of:

converting the message from an original format to a standard format prior to

15   storing a copy onto the persistent storage device, wherein the stored copy is in the standard format; and

converting the message from the standard format into the original format just before providing it to the receiving process.
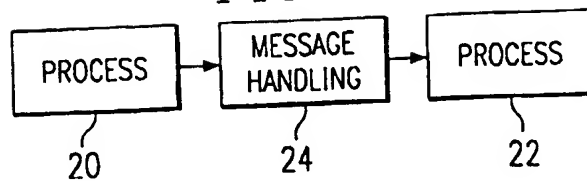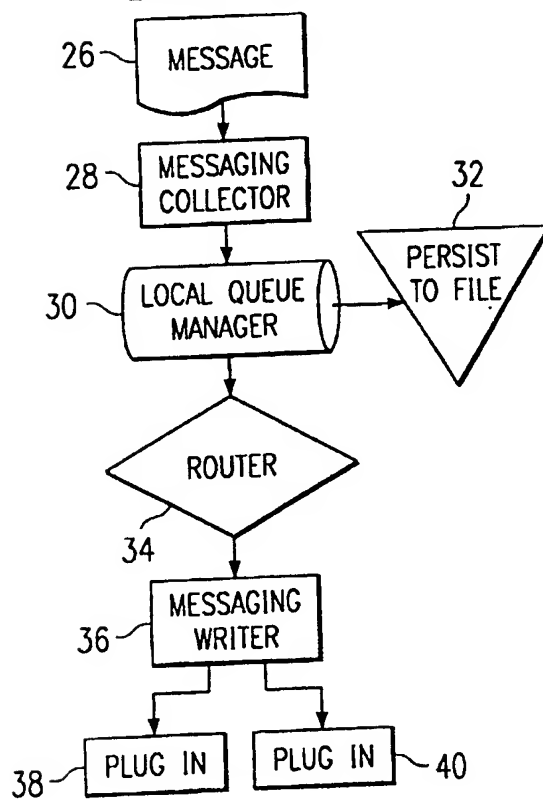
FIG. 1

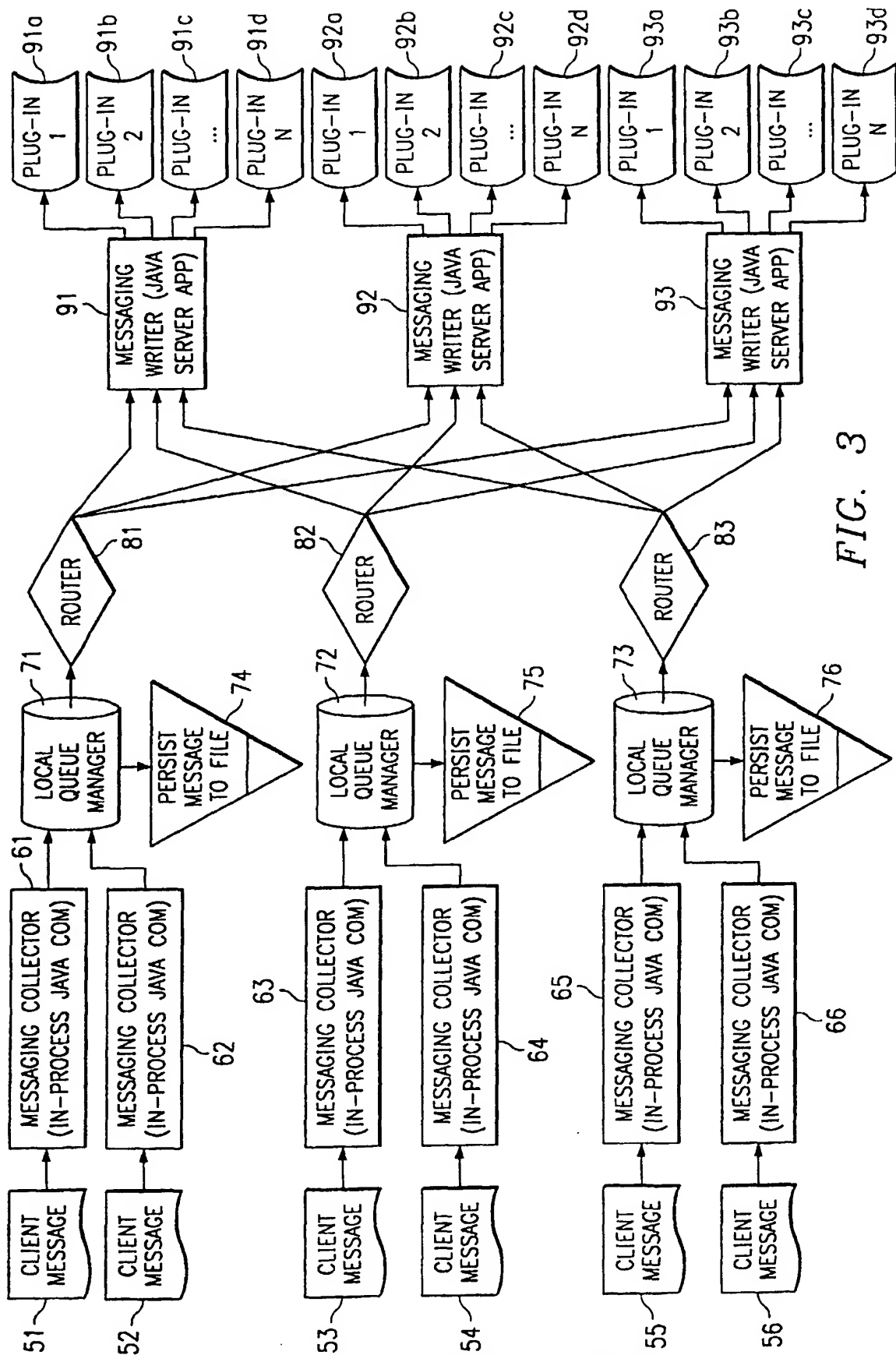PROCESS — MESSAGE HANDLING — PROCESS

20   24   22

FIG. 2

26 — MESSAGE

28 — MESSAGING COLLECTOR

30 — LOCAL QUEUE MANAGER → PERSIST TO FILE   32

ROUTER

34

36 — MESSAGING WRITER

38 — PLUG IN   PLUG IN — 40

FIG. 3

INTERNATIONAL SEARCH REPORT

A.	CLASSIFICATION OF SUBJECT MATTER
IPC(7)	: G06F 15/16, H04L 12/00
US CL	: 709/206, 709/232, 709/245

According to International Patent Classification (IPC) or to both national classification and IPC

B.	FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
    U.S. : 709/206, 709/232, 709/245

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
None

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
East

C.	DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 6,009,462 A (BIRRELL et al) 28 December 1999, column 1, lines 64-67; column 2, lines 1-17.	1-7
Y	US 6,014,711 A (BROWN) 11 January 2000, column 1, lines 6-13.	1-7
Y, P	US 6,092,114 A (SHAFFER et al) 18 July 2000, column 3, lines 51-64.	1-7
Y, E	US 6,212,552 B1 (BILIRIS et al) 03 April 2001, column 3, lines 15-25.	1-7

☐ Further documents are listed in the continuation of Box C.   ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance		
"E"	earlier application or patent published on or after the international filing date	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed	"&"	document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
14 May 2001 (14.05.2001)	0 8 JUN 2001
Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Meng-Ai An Peggy Harrod
Facsimile No. (703)305-3230	Telephone No. (703)305-3900

Form PCT/ISA/210 (second sheet) (July 1998)